

DROWSINESS DETECTION

- **## 68-Landmark Predictor**
- **## dlib library's shape_predictor**
 - Loading Face Detector & Landmark Predictor
 - Reading Frame from Camera
 - Converting frame from BGR to Gray
 - Detecting Face
 - Identifying Left & Right Eye Coordinates
 - Finding Eye Aspect Ratio
 - Eye close detection

In []:

```
from scipy.spatial import distance as dist #Mathematical Operations - TYo calculate Distance
from imutils import face_utils #For coordinates
import imutils #For Resizing
import dlib #To Load 68-LandMark files
import cv2
import winsound #For Beep Sound (NO installation required)

frequency = 2500 #Sound Frequency
duration = 1000 #Duration of Sound (Duration - 1000 = 1 - Second)

def eyeAspectRatio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

# ear - eye aspect ratio

count = 0
earThresh = 0.3 #Distance between vertical eye coordinate Threshold
earFrames = 48 #Consecutive frames for Eye closure
shapePredictor = "D:/shape_predictor_68_face_landmarks.dat"

cam = cv2.VideoCapture(2)
detector = dlib.get_frontal_face_detector() #To detect face coordinates - inbuilt Libraries.
predictor = dlib.shape_predictor(shapePredictor) #ShapePredictor file

# Get the Coordinates of Left & Right Eye
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

while True:
    _, frame = cam.read()
    frame = imutils.resize(frame, width = 450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    rects = detector(gray, 0)

    for rect in rects:
        shape = predictor(gray, rect) #parameter - gray & rect : On Gray image we apply rectangle.
        shape = face_utils.shape_to_np(shape) # Convert into Array to extract the coordinates for eye

        leftEye = shape[lStart:lEnd] #LeftEye - Start point to End point
```

```

rightEye = shape[rStart:rEnd] #RightEye - Start point to End point
leftEAR = eyeAspectRatio(leftEye) # Coordinates for Left Eye
rightEAR = eyeAspectRatio(rightEye) # Coordinates for Right Eye

ear = (leftEAR + rightEAR) / 2.0

leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [leftEyeHull], -1, (0, 0, 255), 1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 0, 255), 1)

if ear < earThresh:
    count += 1

    if count >= earFrames:
        cv2.putText(frame, "DROWSINESS DETECTED", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        windsound.Beep(frequency, duration)
    else:
        count = 0

cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

if key == ord("q"):
    break

cam.release()
cv2.destroyAllWindows()

```